Reduced Order Modelling of Aircraft Gust Response for Use in Early Design Stages

Stephen P. I. Williams[1], Dorian P. Jones[2], Ann L. Gaitonde[3], Chris Wales[4] and Samantha J. Huntley[5]
Department of Aerospace Engineering, University of Bristol, Bristol, England, BS8 1TR

This paper expands on the work previous carried out by authors in exploring the role Reduced Order Models (ROMs) can play in the early stages of the aircraft design cycle, particularly in respect of modelling the response to atmospheric gusts. Gusts have a large impact on the development of aircraft due to their prevalence among the critical design cases for loads. Despite this, the traditional methods of modelling such cases are restricted due to high costs associated with both wind tunnel testing and Computational Fluid Dynamics (CFD) simulation. This is particularly problematic and restrictive during those early design stages where designs may change significantly and frequently. ROMs have been shown to be capable of reproducing gust responses at lower costs compared to the high fidelity simulations, and previous work presented by the authors introduced a number of techniques that could be used to make the process of gust ROM creation more efficient without loss of accuracy. This paper explores the strengths and weaknesses of these approaches in more detail and introduces further new techniques to improve efficiency. In particular this paper will consider more complex test cases which better reflect the industrial demands upon such a tool which is likely to be used in the design of aircraft.

Nomenclature

$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$	=	system matrices
$\widetilde{\boldsymbol{A}}$, $\widetilde{\boldsymbol{B}}$, $\widetilde{\boldsymbol{C}}$, $\widetilde{\boldsymbol{D}}$	=	discrete-time system matrices
C_l, C_m	=	lift and pitching moment coefficients
$\boldsymbol{H}(t)$	=	continuous-time impulse response matrix
$\boldsymbol{H_{rs}}(k)$	=	Hankel matrix of size r × s at time level k
$\boldsymbol{H_k}$	=	Markov parameter
h_1	=	first order kernel of the system
k	=	time level
l_g	=	gust length
m	=	number of inputs in a multiple input/output system
p	=	number of outputs in a multiple input/output system
t	=	time
U	=	velocity
$\boldsymbol{U}$, $\boldsymbol{V}$	=	square unitary matrices
$v_g(t)$	=	input vector
x	=	state vector
$\dot{x}$	=	state vector differentiated with respect to time
y	=	output vector
y_1	=	non-linear system response to a pulse input of arbitrary
y_{11}	=	non-linear system response to a pulse input of twice the magnitude as that of y_1
$\boldsymbol{0_m}$	=	square identity matrix of size m × m

[1] Ph.D Researcher, Department of Aerospace Engineering, Queens Building, University Walk, Bristol, England, BS8 1TR.
[2] Senior Lecturer, Department of Aerospace Engineering.
[3] Senior Lecturer, Department of Aerospace Engineering.
[4] Research Associate, Department of Aerospace Engineering.
[5] Research Associate, Department of Aerospace Engineering.

$\boldsymbol{0_p}$ = square identity matrix of size p × p
Δt = discrete time-step
$\boldsymbol{\Sigma}$ = a diagonal matrix containing non-negative, real numbers
λ = eigenvalue

I. Introduction

During their operational life, aircraft will encounter numerous gusts; localised and temporary fluctuations in airflow. These gusts have a large influence in design stages as they are typically responsible for many of the critical design cases; due to the effect they have on both loads and fatigue [1]. As such, being able to prove that a given aircraft design can adequately handle the impact of such gusts is heavily regulated as demonstrated in the *Certification Specifications for Large Aeroplanes - CS-25* [2]. These regulations specify a range of gust conditions, both continuous and discrete, for which the design must be shown to withstand without issue. The work in this paper focuses on the discrete form of gust, and specifically those acting in a vertical direction with respect to the aircraft motion.

Whilst gusts can come in a wide range of shapes (defined as the way in which the gust velocity varies as an aircraft travels through it) this paper focuses on the "1-cosine" type gusts laid out in the CS-25 Regulations [2] and sharp-edged gusts. For certification purposes, the CS-25 Regulations [2]) require a suitable number of "1-cosine" gusts of various lengths, ranging from 9m up to 107m (with specifications on additional, long gust lengths if the aircraft being certified is over a certain size) to prove comprehensively that the aircraft is fit for purpose in such conditions. The peak velocity of "1-cosine" gusts are typically taken to comply with these regulations; which use the gust lengths and flight altitude to calculate said velocity.

Despite the importance these aerodynamic gust cases have on the design of aircraft, particularly in the early stages, they represent an area of large, ongoing development as improved methods of simulation are introduced. Their complexity results in large costs, both financially and in terms of resource-hours, both for modelling within wind tunnels, and via Computational Fluid Dynamics (CFD). For wind tunnels these costs are associated with the specialized setup required to generate gusts; typically these setups employ two vanes [3] (aerofoil shaped devices upwind of the model being tested) which oscillate to generate a gust which propagates through the wind tunnel. For CFD, the costs are linked to the computational resources required to carry out simulation on a sufficiently fine mesh over the entire domain to avoid the dissipation of the gust [4]; this often leads to simplifications being made to reduce this cost at the expense of the accuracy of the results. New techniques such as Split Velocity Method (SVM) [5] allows the gust to propagate without the need for a refined far-field mesh. However, CFD is still computationally expensive; making any reduction in the amount of simulation required beneficial. Reduced Order Models (ROMs) created using CFD results have been shown to be capable of reproducing gust responses at lower costs compared to standard, high fidelity simulations [4] and so make the possibility of obtaining accurate gust response results feasible during early design phases.

This paper takes the CFD based ROM originally created by Wales *et al* [4] and further developed by Williams *et al* [6] and both looks at how well the ROM works for more complex test cases, as well as looking at improvements that can be made to efficiency without compromising accuracy. For the former, the ROM is applied to half-aircraft models in viscous, aeroelastic cases, and also to a simpler, full aircraft model within inviscid, rigid cases but with the inclusion of flight mechanics. The accuracy of the ROM is examined and areas of strengths and weaknesses identified and, particularly in areas of less satisfactory accuracy, the probable causes explored. Improvements to the method by which the ROM is built are proposed, with the new methods tested and results examined. In all cases, the ultimate application of the ROM as a design tool is kept at the forefront of all analysis.

II. Summary of the Original Reduced Order Model

The original ROM was built using the methods put forward by Wales *et al* [4] and used by Williams *et al* [6]. As the mathematics behind this ROM is built upon within this paper, the method is summarised below for completeness.

A. Eigensystem Realisation Algorithm for Continuous Systems

This ERA approach views the CFD as an input/output system, where there are a small number of inputs to the system (here simply the vertical gust velocity). Typically the number of outputs from the CFD is large (flow variable values throughout the mesh) and if full flow field data is required no reduction in the size of the problem can be made. However if only a much smaller set of outputs is required such as surface pressures, or as in the cases considered here the lift and pitching moment coefficients, then it is possible to construct a ROM of this input-output

system using ERA. Then if the non-linear state flow can be assumed to be close to linear during the gust encounter, then for a continuous time system, the behaviour of said system can be expressed as:

$$\begin{aligned} \dot{x}(t) &= \boldsymbol{A}x(t) + \boldsymbol{B}v_g(t) \\ y(t) &= \boldsymbol{C}x(t) + \boldsymbol{D}v_g(t) \end{aligned} \tag{1}$$

Where $v_g(t)$ is the input vector and the output vector contains changes to the lift and moment coefficients:

$$y = \begin{bmatrix} C_l \\ C_m \end{bmatrix} \tag{2}$$

Now it would be possible to generate a continuous time ROM via the Eigensystem Realisation Algorithm (ERA) [7] if the continuous-time impulse response matrix could be identified [4]. However these are not in general available as the majority of CFD codes/solvers work in discretised time rather than continuous time. Thus it is necessary to consider the application of this approach in discrete time.

B. Eigensystem Realisation Algorithm for Discrete Systems

The first step in the method implemented here is to form a discrete approximation to Eq. (1) using a first order implicit finite difference scheme. On rearranging [4] this becomes:

$$\begin{aligned} \tilde{x}_k &= \widetilde{\boldsymbol{A}}\tilde{x}_{k-1} + \widetilde{\boldsymbol{B}}\left(\tilde{v}_g\right)_k \\ \tilde{y}_k &= \widetilde{\boldsymbol{C}}\tilde{x}_k + \widetilde{\boldsymbol{D}}\left(\tilde{v}_g\right)_k \end{aligned} \tag{3}$$

Where subscript k represents the discrete time level and the discrete system matrices are related to the continuous system matrices as follows:

$$\begin{aligned} \widetilde{\boldsymbol{A}} &= (I - \boldsymbol{A}\Delta t)^{-1} \\ \widetilde{\boldsymbol{B}} &= (I - \boldsymbol{A}\Delta t)^{-1}\boldsymbol{B}\Delta t \\ \widetilde{\boldsymbol{C}} &= \boldsymbol{C} \\ \widetilde{\boldsymbol{D}} &= \boldsymbol{D} \end{aligned} \tag{4}$$

Note that the title is used to identify values, vectors & matrices in discrete time. In most test cases the system matrix $\boldsymbol{D}$ is both small and known (for gust responses it is simply zero), thus the output vector (Eq. (3)) can be modified to $\tilde{y}^m{}_k = \boldsymbol{C}\tilde{x}_k$ for easier implementation. Then the solution in matrix form to the problem where the system is initially undisturbed can be shown to be [4]:

$$\tilde{y}_l^m = [\boldsymbol{H_l}, \quad \boldsymbol{H_{l-1}}, \quad \dots, \quad \boldsymbol{H_2}, \quad \boldsymbol{H_1}, \quad \boldsymbol{H_0}] \begin{bmatrix} \left(\tilde{v}_g\right)_0 \\ \left(\tilde{v}_g\right)_1 \\ \vdots \\ \left(\tilde{v}_g\right)_{l-2} \\ \left(\tilde{v}_g\right)_{l-1} \\ \left(\tilde{v}_g\right)_l \end{bmatrix} \tag{5}$$

Where the terms in the following sequence $\boldsymbol{H}_k$, $k = 0, \infty$ are the discrete Markov parameters of the linear system:

$$\{\boldsymbol{H}_0, \quad \boldsymbol{H_1}, \quad \dots \quad \boldsymbol{H}_k, \quad \dots\} = \{\widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{B}}, \quad \widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{B}}, \quad \dots, \quad \widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}^k\widetilde{\boldsymbol{B}}, \quad \dots\} \tag{6}$$

and the sequence is called the impulse-response, the weighting sequence or the Markov sequence of the system. The Markov parameters can be directly identified from the responses to discrete impulses on each input channel of the discrete linear system, which here would be the dynamically linearised Euler & Navier-Stokes equations, and for a system with m inputs and p outputs, each Markov parameter is a $p \times m$ matrix.

C. Pulse Extraction

If a linearized version of the non-linear CFD code is not available, an alternative approach to obtain approximate linear responses from a non-linear CFD code is available using Volterra theory [8–10]. A truncated Volterra series for small inputs can be used to represent the solution for weakly non-linear systems. Should the system be able to be approximated by a second order Volterra series, then the first order kernel of the system can be identified as:

$$h_1 = 2y_1 - \frac{1}{2}y_{11} \tag{7}$$

h_1 captures some level of the amplitude dependence for non-linear systems, thus is generally different from the purely linear pulse response. However, should the non-linear system exhibit near linear behaviour in response to small inputs, then it can be assumed that the non-linear first order kernel is that of an approximating linear system. Ergo the linear responses can be found from two response of the non-linear code.

D. System Reduction

System reduction is achieved using ERA, which is described in detail in a number of references [4,7,11] hence only the form of the reduced order system matrices is presented here. There are several possible realisations for the reduced system matrices and in this work a balanced realisation is used given by [7]:

$$\begin{aligned} \widetilde{\boldsymbol{A}}_r &= \boldsymbol{\Sigma}_{\mathrm{r}}^{-\frac{1}{2}}\, \boldsymbol{U}_r^T \boldsymbol{H}_{rs}(1) \boldsymbol{V}_r \boldsymbol{\Sigma}_{\mathrm{r}}^{-\frac{1}{2}} \\ \widetilde{\boldsymbol{B}}_r &= \boldsymbol{\Sigma}_{\mathrm{r}}^{\frac{1}{2}}\, \boldsymbol{V}_r^T \boldsymbol{E}_m \\ \widetilde{\boldsymbol{C}}_r &= \boldsymbol{E}_p^T \boldsymbol{U}_r \boldsymbol{\Sigma}_{\mathrm{r}}^{\frac{1}{2}} \end{aligned} \tag{8}$$

Where $\boldsymbol{H}_{rs}(k)$ is a Hankel matrix (made up of Markov parameters) [4] and:

$$\boldsymbol{H}_{rs}(0) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \approx \boldsymbol{U}_r \boldsymbol{\Sigma}_{\mathrm{r}} \boldsymbol{V}_r^T \tag{9}$$

$$\begin{aligned} \boldsymbol{E}_m^T &= [\mathbf{I}_{\mathrm{m}},\ \ \mathbf{0}_{\mathrm{m}},\ \ \mathbf{0}_{\mathrm{m}},\ \ \ldots,\ \ \mathbf{0}_{\mathrm{m}}] \\ \boldsymbol{E}_p^T &= [\mathbf{I}_{\mathrm{p}},\ \ \mathbf{0}_{\mathrm{p}},\ \ \mathbf{0}_{\mathrm{p}},\ \ \ldots,\ \ \mathbf{0}_{p}] \end{aligned} \tag{10}$$

Here $\mathbf{I_m}$ and $\mathbf{I_p}$ are identity matrices of size m×m and p×p respectively, and $\mathbf{0_m}$ and $\mathbf{0_p}$ are zero matrices of size m×m and p×p respectively. The entries in the Hankel matrices are simply the system Markov parameters found via the response of the system to impulse inputs. For a gust the pulse input required would be of the form $(\tilde{v}_g)_0 = 1$, $(\tilde{v}_g)_k = 0\,,\ k \geq 1$. However, this is not a very efficient approach since as explained by Wales *et al* [12] who states that "*the size of the effective pulse, which depends on the time step, relative to the mesh cells in the far field can mean that the change in vertical velocity may not impact on any cell centre*". Thus the same approach as proposed by Wales *et al* [12] to obtain the pulse responses indirectly using discrete differences to find the pulse from the sharp edge gust ('step-up') response is implemented here.

E. System Restarting

To ensure the ROM built is stable, a technique known as restarting is used. The technique used, as demonstrated for ERA by Wales *et al* [13], takes the eigenvalues produced by the ROM and checks if any of the eigenvalues have a magnitude greater than 1 (for a discrete system). If any do, then the system is deemed unstable and restarting is carried out; otherwise the system is stable and the ROM is produced. If restarting is carried out, a polynomial transformation is used where the roots are placed where the unstable eigenvalues occur. This allows a new set of eigenvalues to be produced, without the previously identified unstable ones present. This method is repeated until the stability criteria are met. The technique demonstrated for ERA by Wales *et al* [13] is utilised within the ROMs used in the work put forward in this paper.

F. Improvements in the Construction of the ROM

Williams *et al* [6] used the aforementioned algorithm to build a ROM; this was used on test cases using a 3-Dimensional wing model for inviscid, rigid simulations. They then modified this ROM to reduce the computational costs whilst largely maintaining the accuracy of the results. These modifications included: the reduction from two

sharp-edge gusts to one; the introduction of variable time-steps to minimize the number of time-steps required to propagate the sharp-edged gust in through the domain to just ahead of the model; and the minimizing of the number of time-steps required after the sharp-edged gust has cleared the model. The final ROM created by Williams *et al* [6] is used as the baseline ROM within this paper, and the work put forward is a continuation upon these modifications.

III. Adaptation to the Hankel Matrix Construction

Williams *et al* [6] explored the restriction of the number of time-steps calculated after the sharp-edge gust (used to create the ROM) has cleared the geometry. They found that, for the test cases considered, the accuracy of the system response was largely maintained until the restrictions got to around 4 chord lengths past the geometry. At this point a sharp degradation in accuracy was observed (leading to the authors only restricting the calculations to 5 chord lengths past the geometry [6]). To help overcome this, it is necessary to modify the way in which the Hankel matrix is constructed. Instead of using a standard 'step-up' response, that is to say the response of the system to a sharp-edged gust, a 'step-down' response can instead be used. This step-down response (Fig. 2) is generated from subtracting the system response to a sharp-edged gust (Fig. 1), away from the steady state of the system.

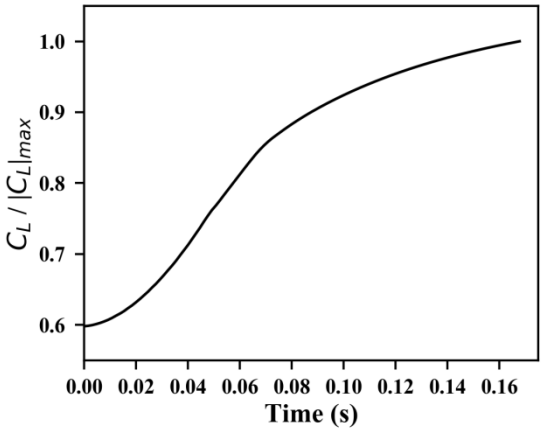

Figure 1. Example of a system response to a sharp-edged gust (a step-up response).

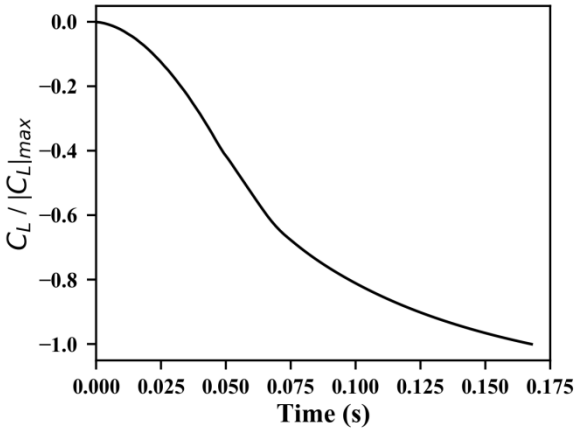

Figure 2. Example of a step-down response, calculated using by subtracting the response to a sharp-edged gust, away from the steady state of the system.

With this new input for the ROM, it is necessary to modify the procedure outlined in section II of this paper. If we take Eq. (1) to Eq. (4) where the initial solution is the steady state, with $\bar{\cdot}$ denoting a steady state value, $\tilde{u} = \overline{u}$ and at the first time-step $\underline{\tilde{u}_1} = \underline{0}$:

$$\underline{\tilde{x}}_1 = \widetilde{\boldsymbol{A}}\underline{x}_0 + \widetilde{\boldsymbol{B}}\underline{\tilde{u}_1} \tag{11}$$

So:

$$\begin{aligned} \underline{\tilde{x}}_1 &= \widetilde{\boldsymbol{A}}\left(-\boldsymbol{A}^{-1}\boldsymbol{B}\overline{\underline{u}}\right) + \widetilde{\boldsymbol{B}}\underline{0} \\ \underline{\tilde{y}}_1 &= \widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}(-\boldsymbol{A}^{-1}\boldsymbol{B}\overline{\underline{u}}) \end{aligned} \tag{12}$$

And:

$$\begin{aligned} \underline{\tilde{x}}_2 &= \widetilde{\boldsymbol{A}}\underline{x}_1 \\ \underline{\tilde{y}}_2 &= \boldsymbol{C}\underline{\tilde{x}}_2 \end{aligned} \tag{13}$$

Then:

$$\begin{aligned} \underline{\tilde{x}}_2 &= \widetilde{\boldsymbol{A}}^2\left(-\boldsymbol{A}^{-1}\boldsymbol{B}\overline{\underline{u}}\right) \\ \underline{\tilde{y}}_2 &= \widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}^2(-\boldsymbol{A}^{-1}\boldsymbol{B}\overline{\underline{u}}) \end{aligned} \tag{14}$$

Therefore the Hankel entries can be written as:

$$\begin{aligned} h_0 &= \widetilde{\boldsymbol{C}}\boldsymbol{A}^{-1}\boldsymbol{B}\underline{\overline{u}} \\ h_1 &= \widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}\boldsymbol{A}^{-1}\boldsymbol{B}\underline{\overline{u}} \\ h_r &= \widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}^{r}\boldsymbol{A}^{-1}\boldsymbol{B}\underline{\overline{u}} \end{aligned} \tag{15}$$

If we replace $-\boldsymbol{A}^{-1}\boldsymbol{B}$ with $\widehat{\boldsymbol{B}}$ then the Hankel matrix becomes:

$$\boldsymbol{H}(0) = \begin{bmatrix} -\widetilde{\boldsymbol{C}}\widehat{\boldsymbol{B}} & -\widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}\widehat{\boldsymbol{B}} & \cdots & -\widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}^{r}\widehat{\boldsymbol{B}} \\ -\widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}\widehat{\boldsymbol{B}} & -\widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}^{2}\widehat{\boldsymbol{B}} & & \vdots \\ \vdots & & \ddots & \vdots \\ -\widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}^{r}\widehat{\boldsymbol{B}} & \ldots & \ldots & -\widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{A}}^{2r+1}\widehat{\boldsymbol{B}} \end{bmatrix} \tag{16}$$

Where:

$$\begin{aligned} \widetilde{\boldsymbol{A}} &= (\boldsymbol{I} - \boldsymbol{A}\Delta t)^{-1} \\ \widehat{\boldsymbol{B}} &= -\boldsymbol{A}^{-1}\boldsymbol{B} \end{aligned} \tag{17}$$

Therefore for the continuous system, the reduced matrices can be written as:

$$\begin{aligned} \boldsymbol{A}_r &= \left(I - \widetilde{\boldsymbol{A}}_r^{\,-1}\right)\frac{1}{\Delta t} \\ \boldsymbol{B}_r &= -\boldsymbol{A}_r\widehat{\boldsymbol{B}}_r \\ \boldsymbol{C}_r &= \widetilde{\boldsymbol{C}}_r \end{aligned} \tag{18}$$

Or, for the discrete system:

$$\widetilde{\boldsymbol{B}} = (\boldsymbol{I} - \boldsymbol{A}\Delta t)^{-1}\boldsymbol{B}\Delta t \tag{19}$$

As:

$$\widehat{\boldsymbol{B}} = -\boldsymbol{A}^{-1}\boldsymbol{B} \tag{20}$$

Then:

$$-\boldsymbol{A}\widehat{\boldsymbol{B}} = \boldsymbol{B} \tag{21}$$

So:

$$(\boldsymbol{I} - \boldsymbol{A}\Delta t)\widetilde{\boldsymbol{B}} = \boldsymbol{B}\Delta t \tag{22}$$

Ergo:

$$\widetilde{\boldsymbol{B}} = -(\boldsymbol{I} - \boldsymbol{A}\Delta t)^{-1}\boldsymbol{A}\widehat{\boldsymbol{B}}\Delta t \tag{23}$$

Therefore:

$$\widetilde{\boldsymbol{B}} = -\widetilde{\boldsymbol{A}}\boldsymbol{A}\widehat{\boldsymbol{B}}\Delta t = -\widetilde{\boldsymbol{A}}\left(\boldsymbol{I} - \widetilde{\boldsymbol{A}}^{-1}\right)\frac{1}{\Delta t}\,\widehat{\boldsymbol{B}}\Delta t = (\boldsymbol{I} - \widetilde{\boldsymbol{A}})\widehat{\boldsymbol{B}} \tag{24}$$

Thus:

$$\widetilde{\boldsymbol{B}}_r = (\boldsymbol{I} - \widetilde{\boldsymbol{A}}_r)\widehat{\boldsymbol{B}}_r \tag{25}$$

Using this modified methodology, a new ROM was created. For both the original ROM and the modified ROM with a sharp-edged gust that ran until 3 chord lengths past the geometry. By comparing the system response (as calculated by the two ROMs) to a CS-25 Regulation [2] "1-cosine" gust of length 214m and of magnitude 14.010m/s, the effects of the new method can be seen as Figs. 3 to 6:

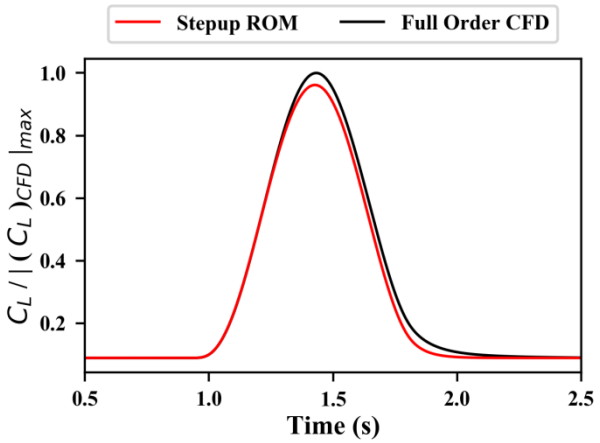

Figure 3. C_L response, for a 214 m long, 14.01 m/s amplitude 1-cosine gust, calculated by the old, step-up Hankel ROM (of size 20).

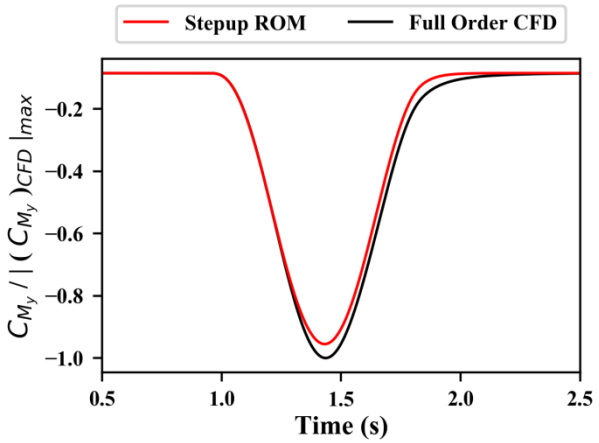

Figure 4. C_{My} response, for a 214 m long, 14.01 m/s amplitude 1-cosine gust, calculated by the old, step-up Hankel ROM (of size 20).

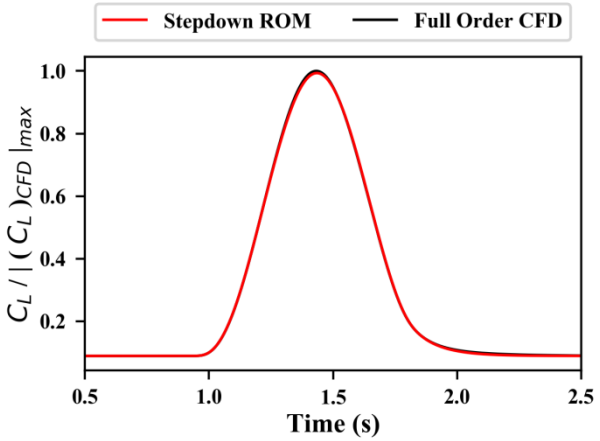

Figure 5. C_L response, for a 214 m long, 14.01 m/s amplitude 1-cosine gust, calculated by the new, step-down Hankel ROM (of size 20).

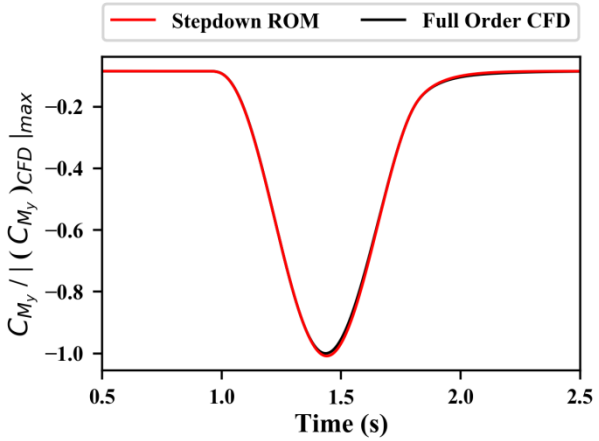

Figure 6. C_{My} response, for a 214 m long, 14.01 m/s amplitude 1-cosine gust, calculated by the new, step-down Hankel ROM (of size 20).

From Figs. 3 to 6 we can see that for both the old and new methods for constructing the Hankel matrix, the initial system response matches the full order CFD result to a satisfactory level. However, once the response gets toward the peak of the gust, the accuracy starts to differ between the two methods. The ROM for the old method under predicts the peak response for the coefficient of lift by a large margin (Fig. 3), whereas the new method is a much closer match, albeit giving a slow over prediction of the peak (Fig. 4); this increased error margin for the old ROM then carries over to the latter part of the response. The increase in accuracy for the new ROM is less evident for the coefficient of pitching moment, although the accuracy is still visibly improved by the new ROM (Fig. 6) compared with the old (Fig. 5).

IV. Improvement of ROM Stability

During testing, it was discovered that a problem exists within the restarting procedure. This problem is intermittent, and appears to be effected by various ROM settings, such as its starting size. When unstable eigenvalues are removed, the restarting method appears to create a pseudo 'blind spot' where further eigenvalues are not identified. This 'blind spot' results in a large shift in the location of the eigenvalues which causes an often extremely large degradation in accuracy is observed; on many occasions this accuracy degradation is to the extreme point whereby the results are meaningless.

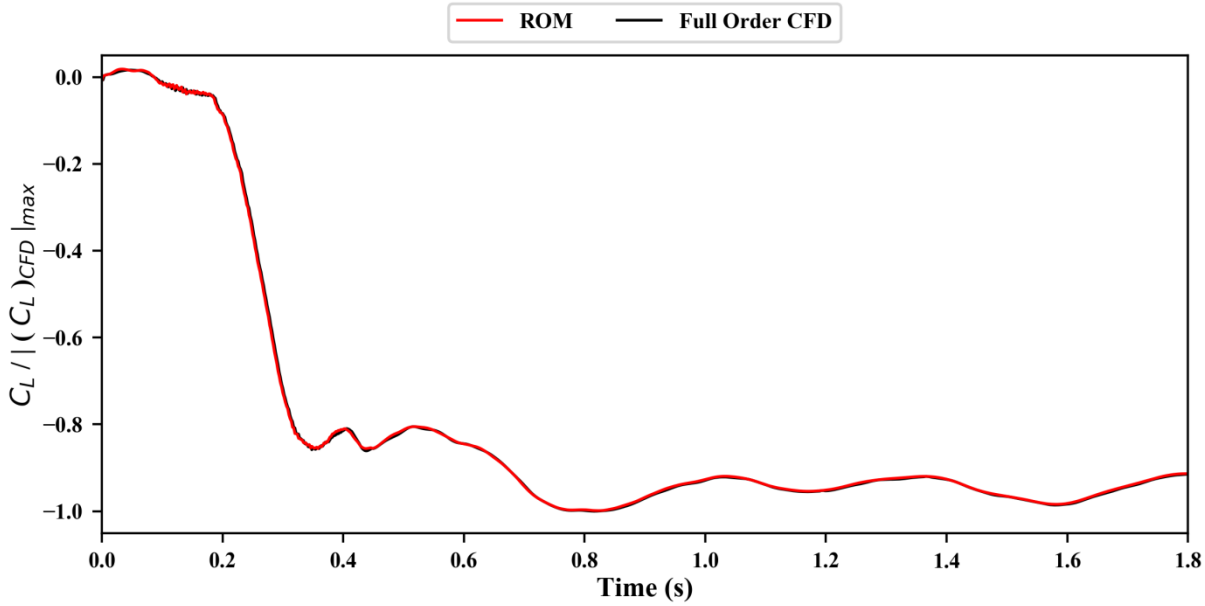

Figure 7. Recreation of a step-down response, for the lift coefficient, by a ROM when the restarting problem isn't noticeably present.

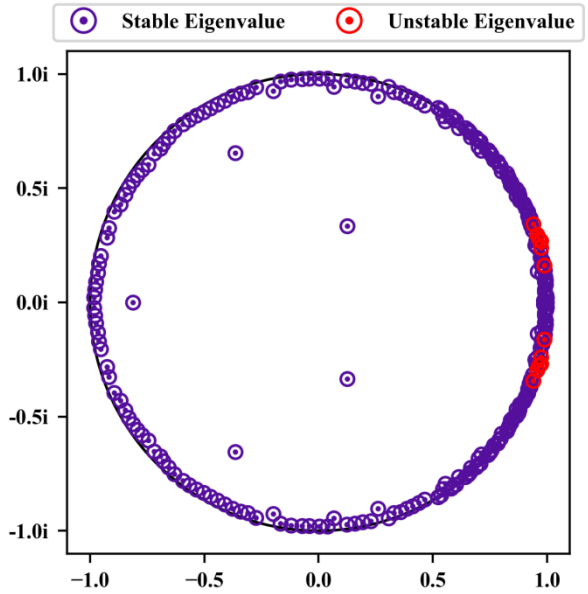

Figure 8. Starting eigenvalues generated by a ROM when the restarting problem isn't noticeably present.

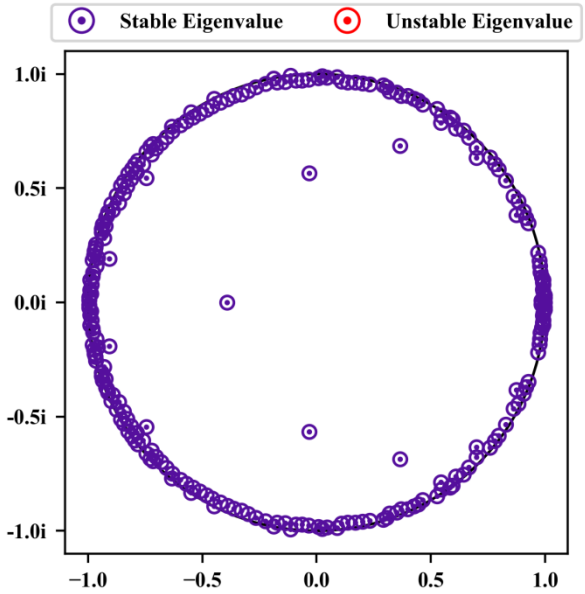

Figure 9. Final eigenvalues generated by a ROM when the restarting problem isn't noticeably present.

Figure 7 shows the recreation of a step-down response (used to create the ROM) using a ROM of size 250 and it can be clearly seen that the system response calculated by the ROM is suitably accurate. Fig. 8 and Fig. 9 show the

first and final set of eigenvalues respectively. Here we can see that the final set of eigenvalues has not noticeably suffered from any issues, with a wise spread of eigenvalues within the stable region (represented by the black circle).

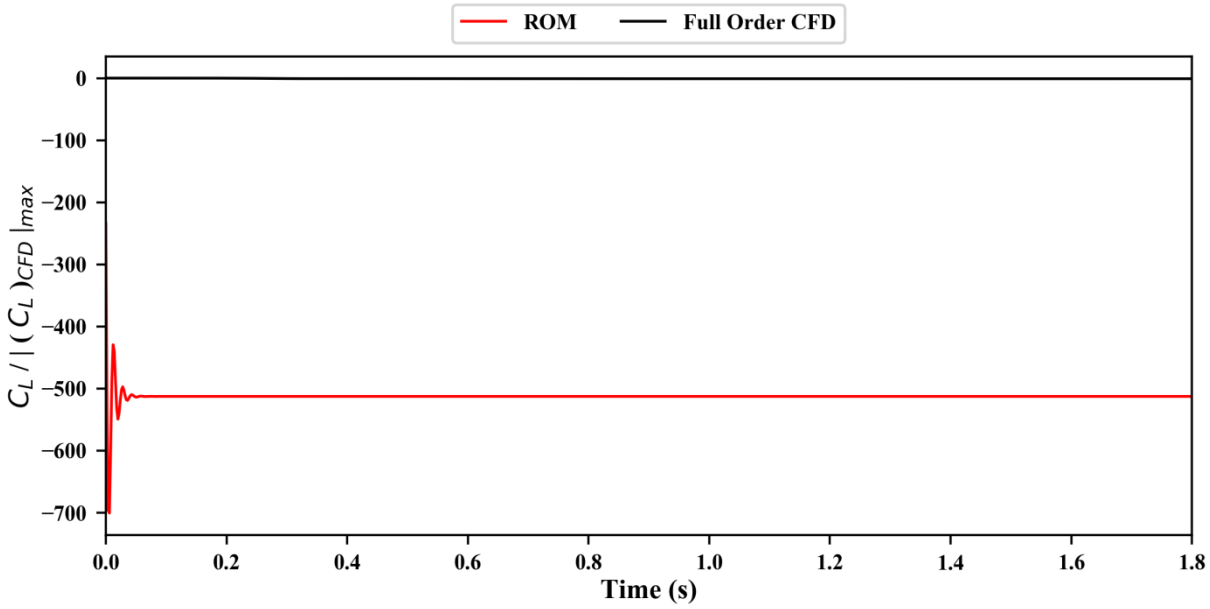

Figure 10. Recreation of a step-down response by a ROM when the restarting problem occurs.

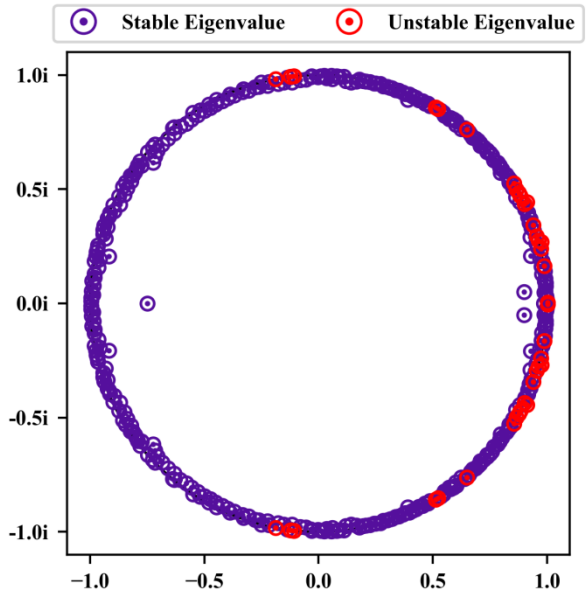

Figure 11. Starting eigenvalues generated by a ROM when the restarting problem occurs.

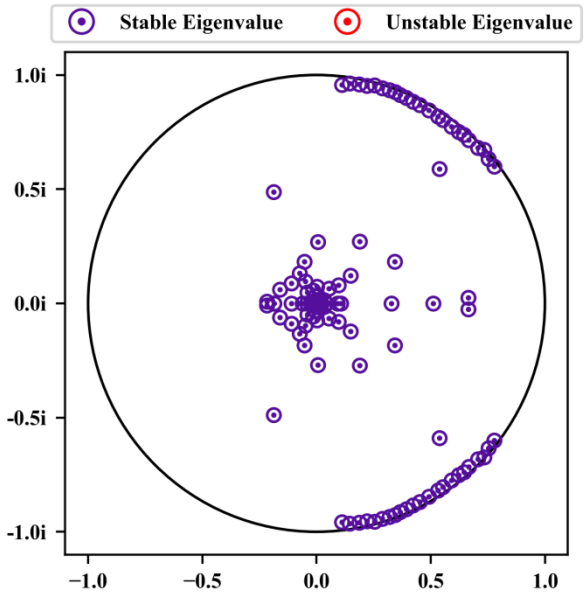

Figure 12. Final eigenvalues generated by a ROM when the restarting problem occurs.

Figure 9 shows the recreation of a step-down response (used to create the ROM) using a ROM of size 400 and it can be clearly seen that the accuracy of the system response has been compromised to a large enough extent as to make the result worthless. Fig. 11 and Fig. 12 show the first and final (after 408 restarts and resultant ROM size reductions) set of eigenvalues respectively and it can be seen that in the final set of eigenvalues, the values have become densely populated around the origin, with no values on the left hand side (toward -1) of the stable region.

What exactly is the cause of this problem remains unknown, with work still being undertaken as to ascertain and correct the apparent limitation in the restarting method. However, it is possible to effectively work around the problem by using a method put forward by McKelvey *et al* [14] which utilises Schur decomposition to allow the

unstable eigenvalues to be projected back within the stable region. Extremely unstable eigenvalues (with an absolute value greater than 2) are set to zero, eigenvalues with an absolute value of exactly 1 have a small value detracted to move them within the stable region, and all other unstable eigenvalues have the Eq. (26) applied [14]:

$$\lambda_{stable} = \lambda_{unstable}\left(\frac{2}{|\lambda_{unstable}|} - 1\right) \tag{26}$$

These, now stable, eigenvalues are then used to reverse the Schur decomposition to produce a stable A matrix.

This technique was then applied to the same case, with a ROM size of 400 (for which the restarting method was unable to produce a meaningful output).

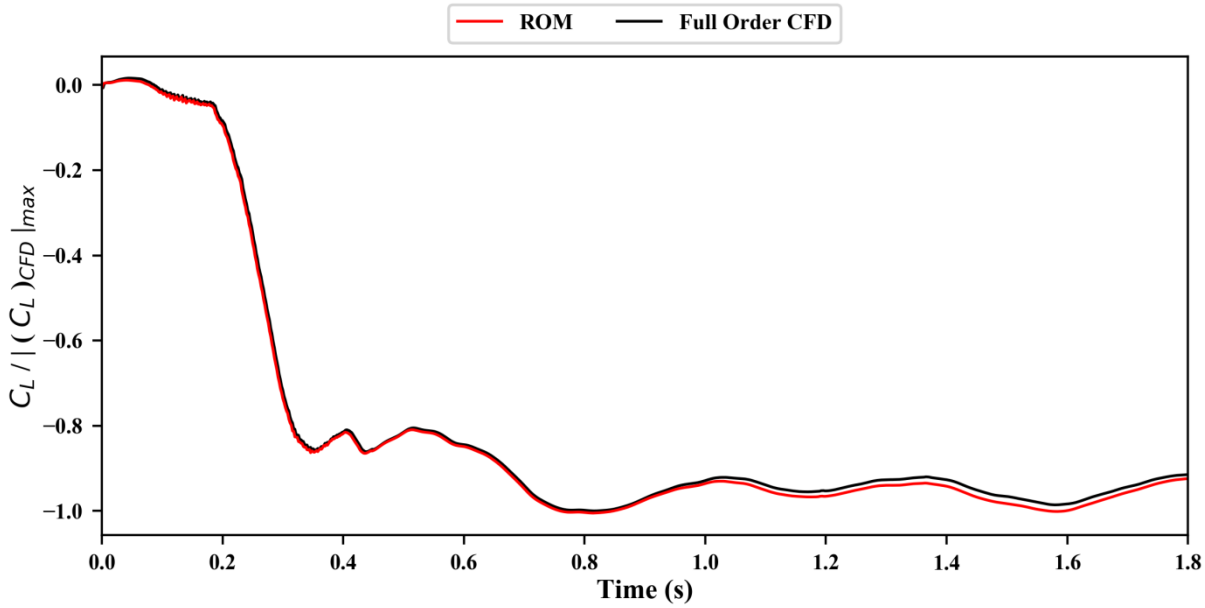

Figure 13. Recreation of a step-down response by a ROM utilising Schur decomposition for stabilisation.

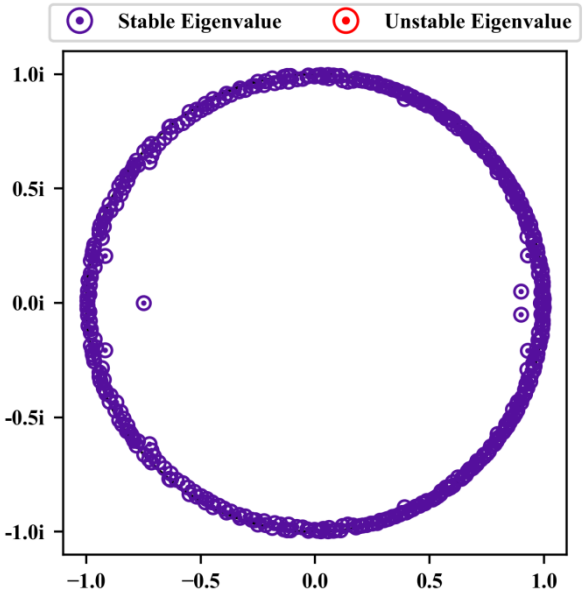

Figure 14. Final eigenvalues generated by a ROM utilising Schur decomposition for stabilisation.

Schur method had the same starting eigenvalues as the restarting method (Fig. 11), but in Fig. 14 we can see that the method results in only the unstable eigenvalues being affected. Figure 13 shows that the Schur decomposition method does produce a relatively stable system, albeit one with a certain degree of inaccuracy starting to appear after about 1 second; although it is worth noting that this error appears to grow rapidly before slightly reducing. This inaccuracy is not instability, but rather the system taking longer to settle; true instability can be seen in Fig. 15, which shows the system response for the same case but without any stabilisation methods being applied.

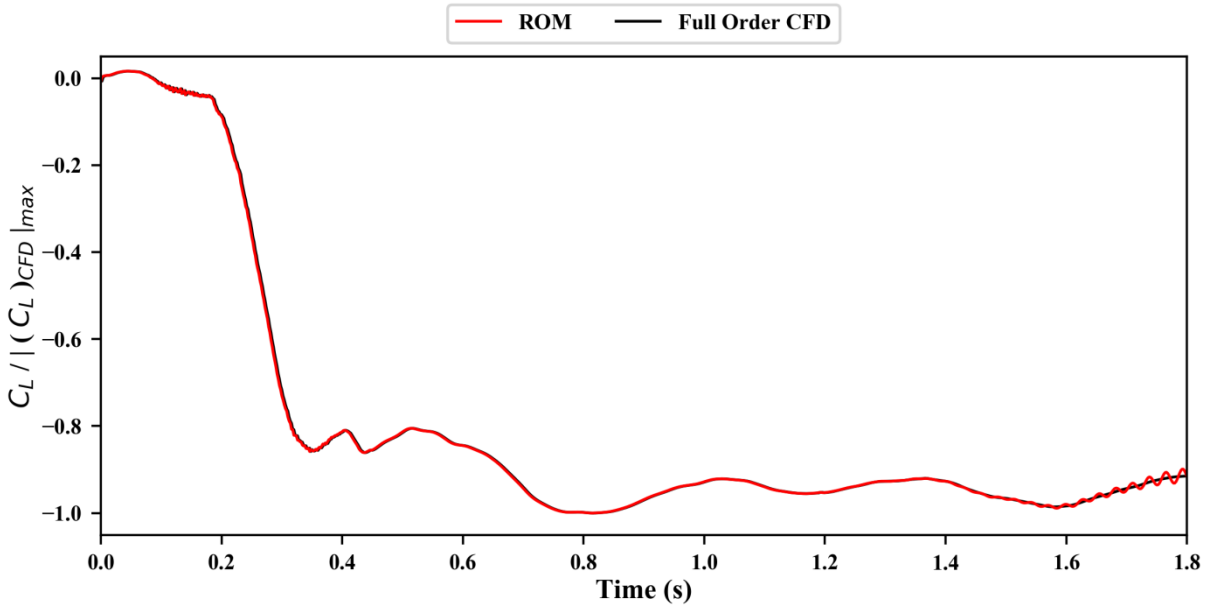

Figure 15. Recreation of a step-down response without any stabilisation.

It can be seen that neither method is without drawback at the current time. The restarting method retrains more information about the system by not manipulating the eigenvalues directly, but it suffers from an inability to find a viable system in some conditions. The Schur decomposition method appears to produce a stable system more reliably, but shows signs of losing some accuracy by directly manipulating the eigenvalues; this was highlighted by the slow settling in Fig. 13. This is therefore an area of ongoing development, with possible corrections to the restarting method being explored, as well as coupling of the two methods.

V. Application of the ROM to Viscous, Aeroelastic Cases

To really explore areas where the ROM works well and where it doesn't, viscous aeroelastic cases were run for a detailed, half aircraft model for a wide range of gusts. The gusts being explored included both a range of gust sizes (Table 2) which were calculated in accordance with the CS-25 regulations [2] and a range of various flight points (Table 1). For all cases, the appropriate gust alleviation factor was employed:

Table 1. Specifications of the four flight points.

Flight point number	Altitude (m)	Mach number
1	0	0.500
2	0	0.800
3	9,142.48	0.800
4	13,106.40	0.800

Table 2. Specifications of the four gust cases.

Gust case number	Gust length (m)	Gust amplitude at sea level (m/s)	Gust amplitude at 9,142.48m (m/s)	Gust amplitude at 13,106.40m (m/s)
1	18	8.798	5.700	4.650
2	80	11.281	7.309	5.963
3	146	12.470	8.080	6.592
4	214	13.291	8.612	7.026

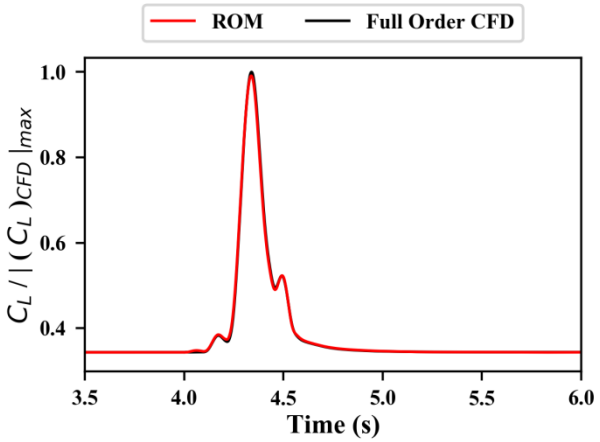

Figure 16. C_L response for a viscous, aeroelastic half-aircraft model for flight point 1, gust case 1.

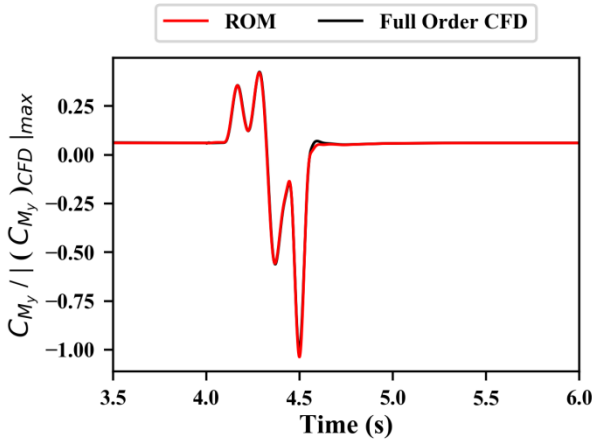

Figure 17. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 1, gust case 1.

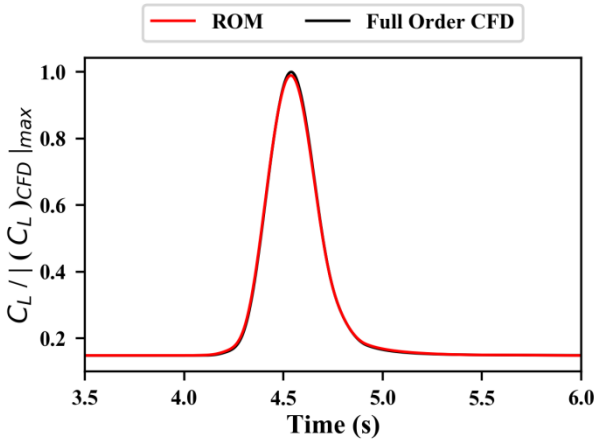

Figure 18. C_L response for a viscous, aeroelastic half-aircraft model for flight point 1, gust case 2.

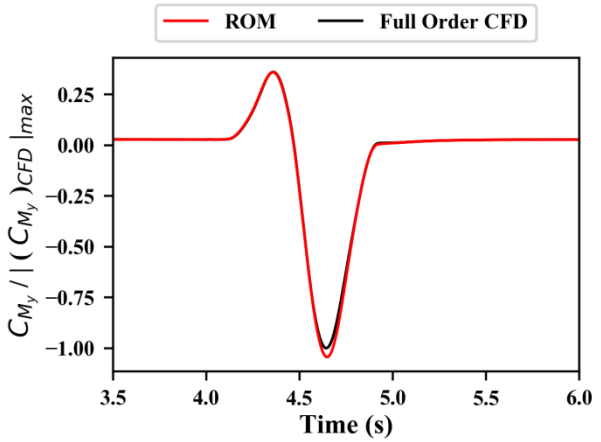

Figure 19. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 1, gust case 2.

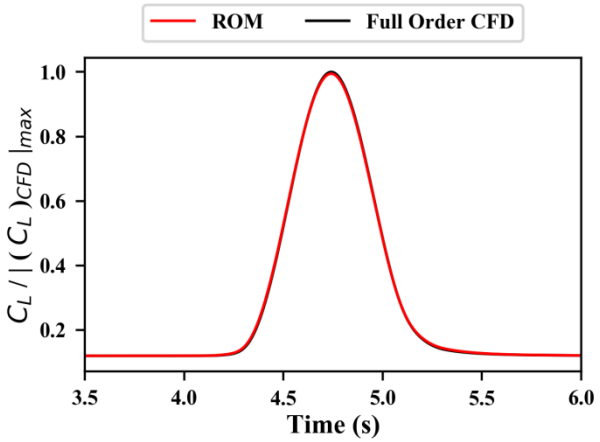

Figure 20. C_L response for a viscous, aeroelastic half-aircraft model for flight point 1, gust case 3.

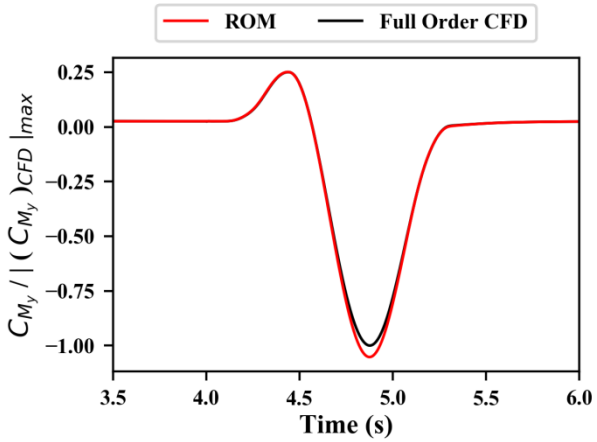

Figure 21. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 1, gust case 3.

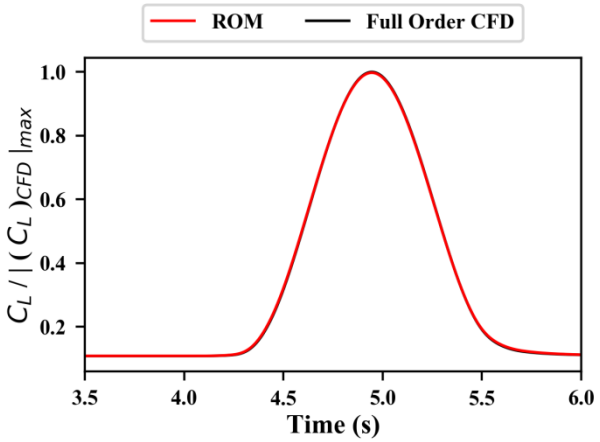

Figure 22. C_L response for a viscous, aeroelastic half-aircraft model for flight point 1, gust case 4.

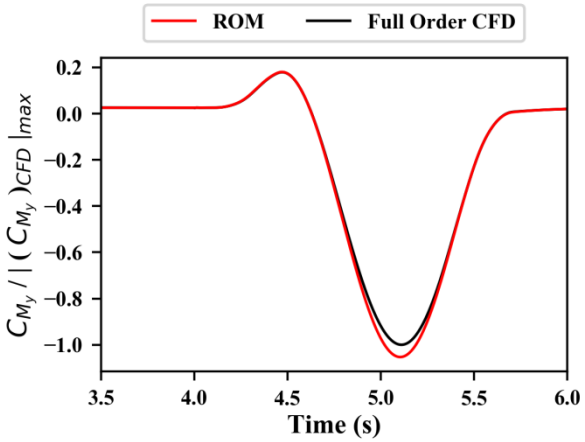

Figure 23. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 1, gust case 4.

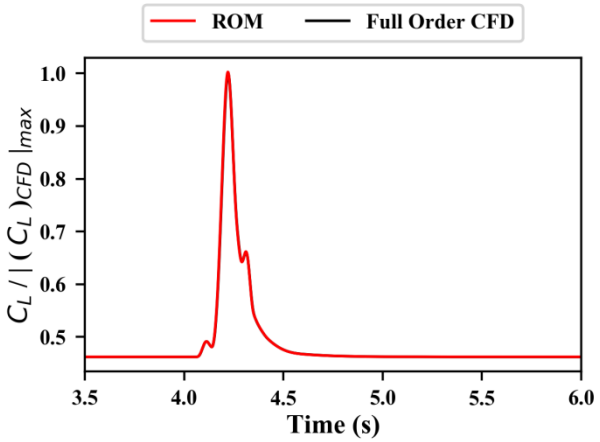

Figure 24. C_L response for a viscous, aeroelastic half-aircraft model for flight point 2, gust case 1.

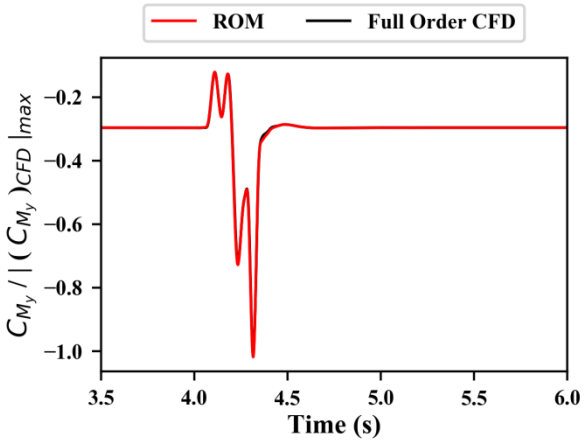

Figure 25. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 2, gust case 1.

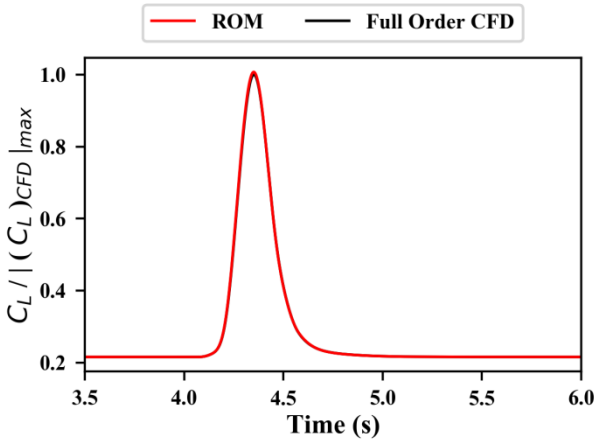

Figure 26. C_L response for a viscous, aeroelastic half-aircraft model for flight point 2, gust case 2.

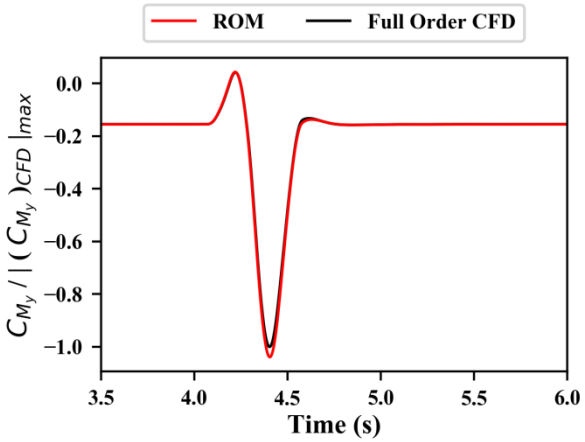

Figure 27. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 2, gust case 2.

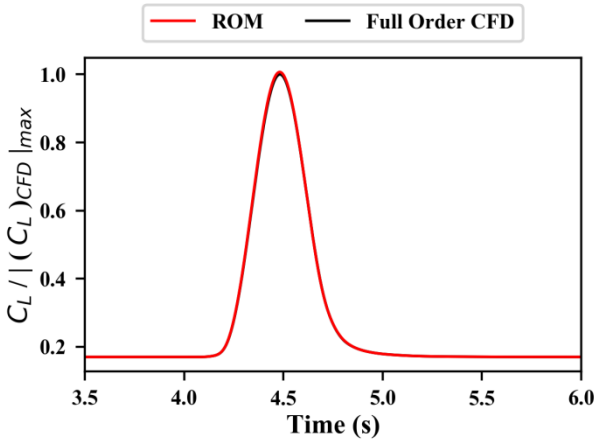

Figure 28. C_L response for a viscous, aeroelastic half-aircraft model for flight point 2, gust case 3.

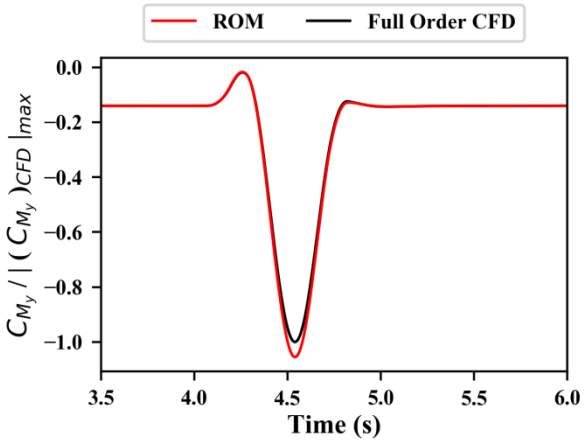

Figure 29. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 2, gust case 3.

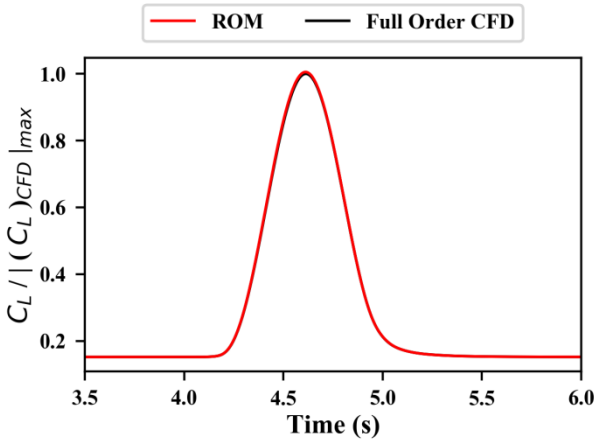

Figure 30. C_L response for a viscous, aeroelastic half-aircraft model for flight point 2, gust case 4.

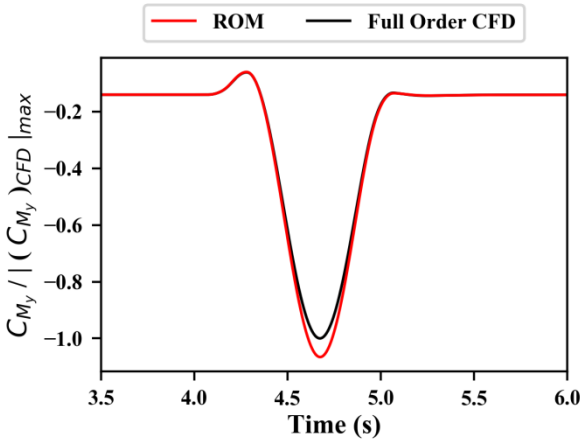

Figure 31. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 2, gust case 4.

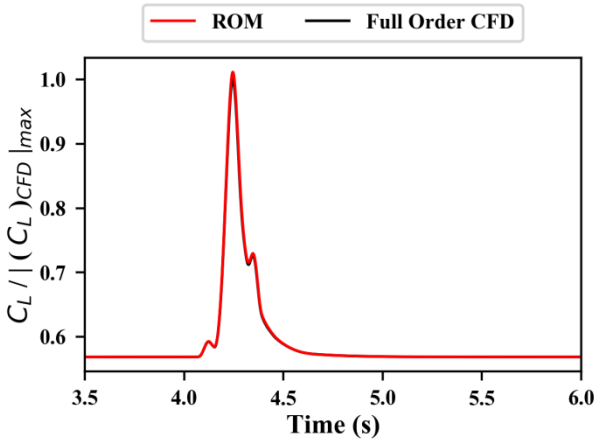

Figure 32. C_L response for a viscous, aeroelastic half-aircraft model for flight point 3, gust case 1.

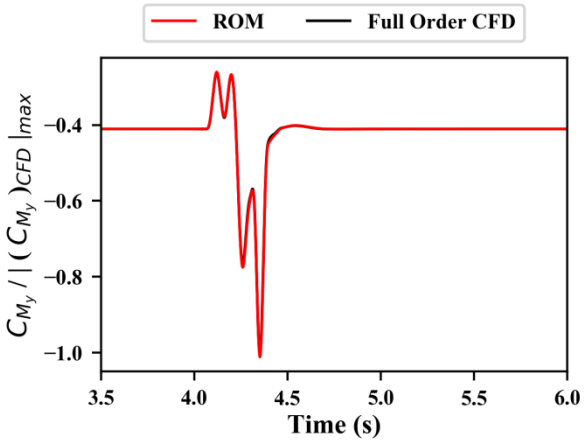

Figure 33. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 3, gust case 1.

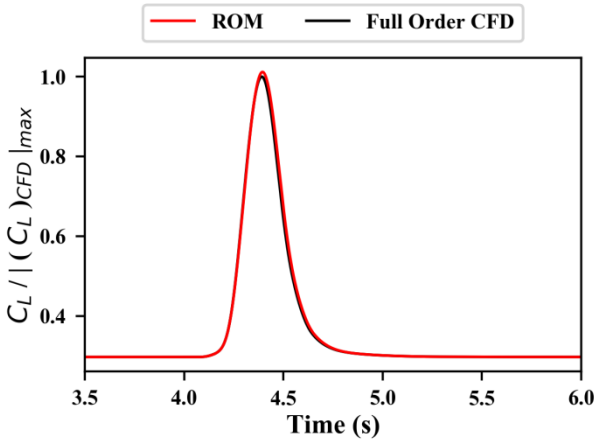

Figure 34. C_L response for a viscous, aeroelastic half-aircraft model for flight point 3, gust case 2.

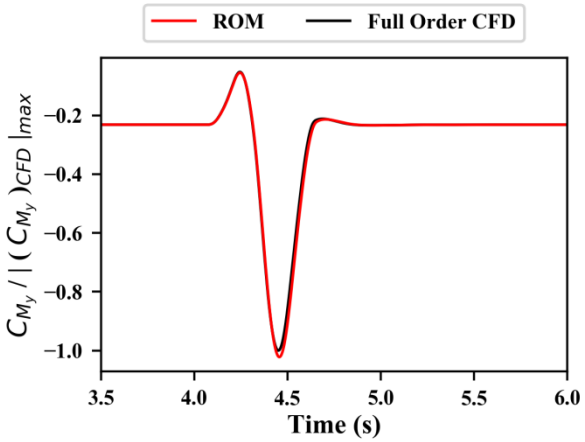

Figure 35. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 3, gust case 2.

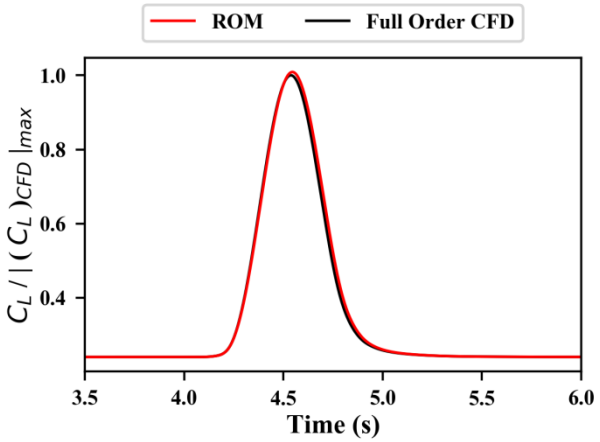

Figure 36. C_L response for a viscous, aeroelastic half-aircraft model for flight point 3, gust case 3.

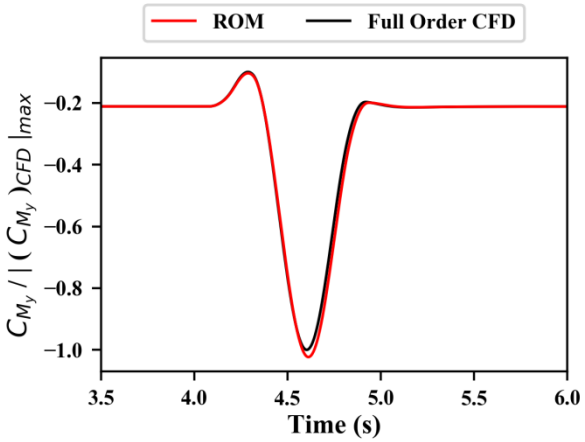

Figure 37. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 3, gust case 3.

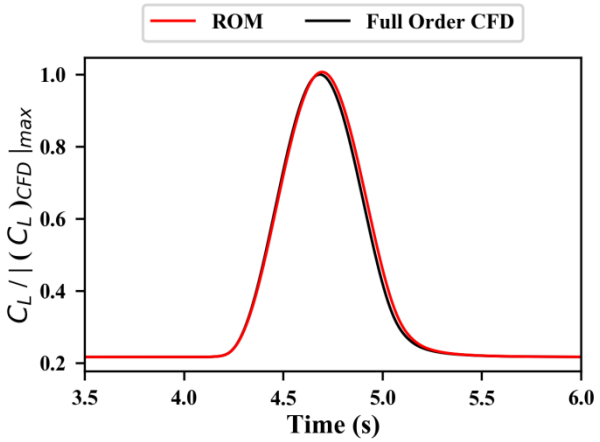

Figure 38. C_L response for a viscous, aeroelastic half-aircraft model for flight point 3, gust case 4.

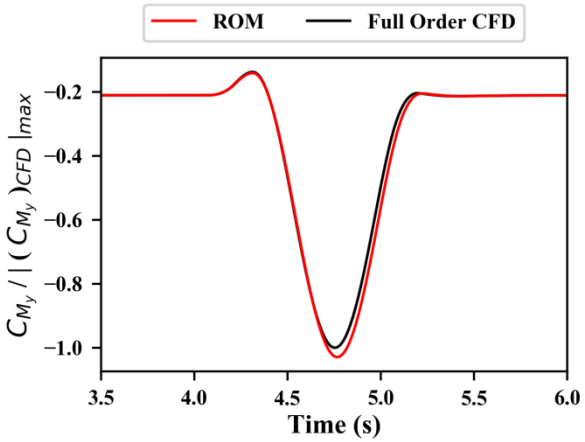

Figure 39. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 3, gust case 4.

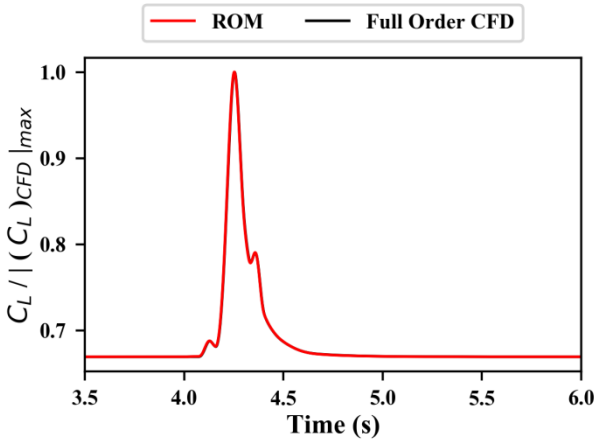

Figure 40. C_L response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 1.

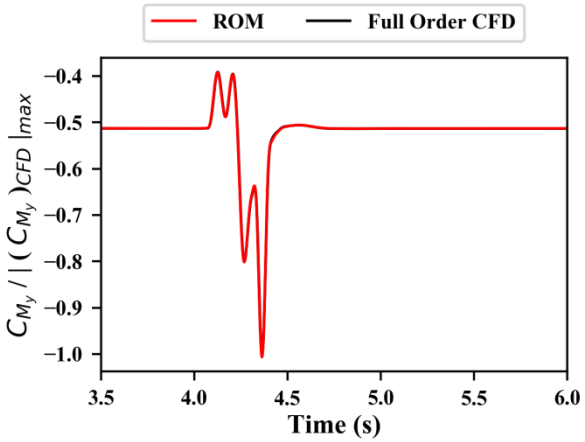

Figure 41. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 1.

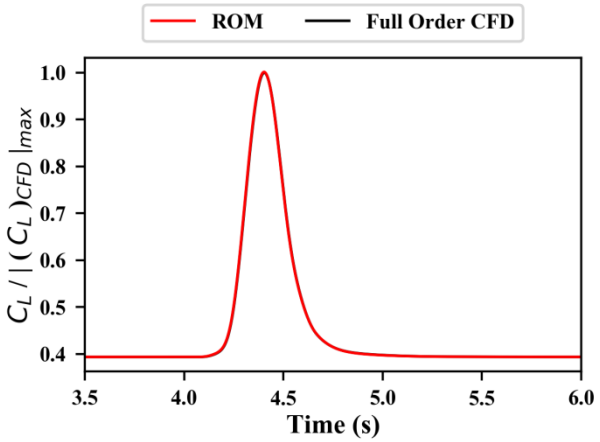

Figure 42. C_L response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 2.

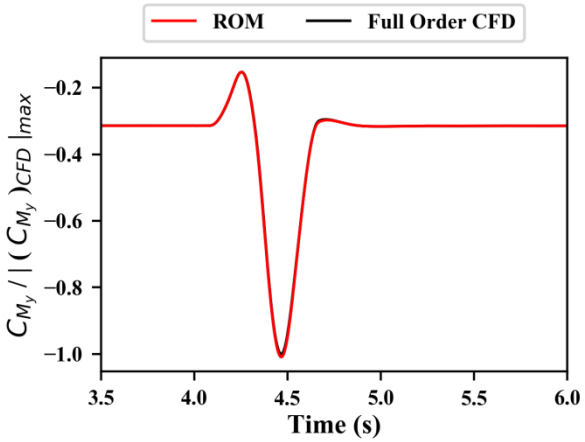

Figure 43. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 2.

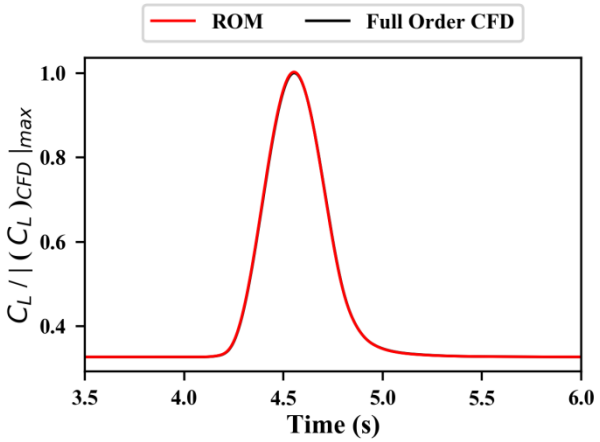

Figure 44. C_L response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 3.

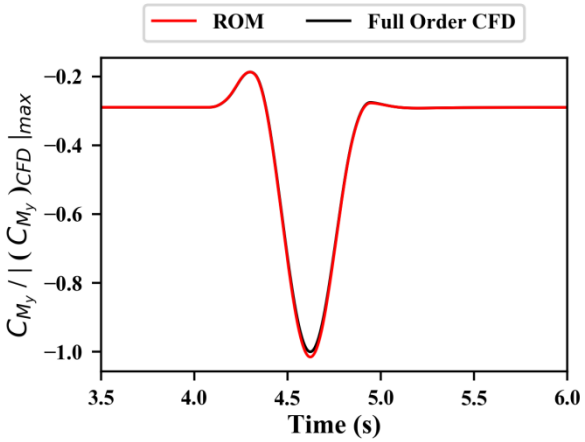

Figure 45. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 3.

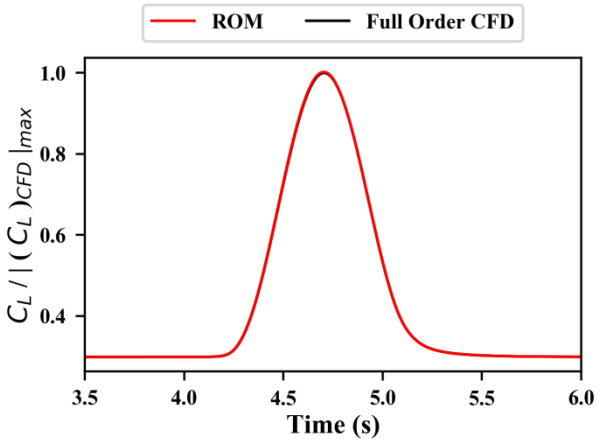

Figure 46. C_L response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 4.

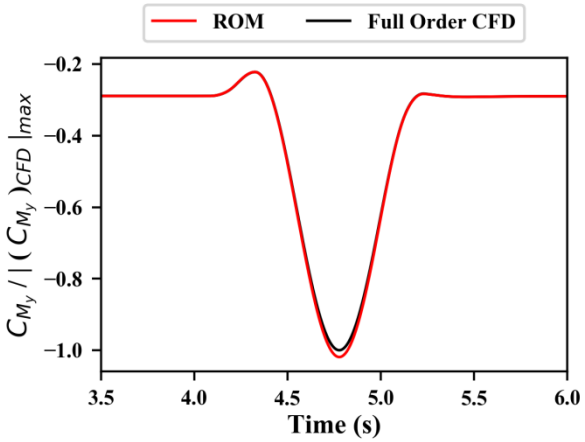

Figure 47. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 4.

It can be seen from Fig. 16 to Fig. 47 that the accuracy of the ROM results is extremely high; with the results for the coefficient of lift in particular often being close to indistinguishable from the full order CFD results. However, it can also be seen that there is a slight degradation in accuracy as the velocity goes up (notably flight point 2) and/or the gust length increases; this is due to the increasing prevalence of non-linearities in the response. However, this degradation in overall accuracy is minor and the level of accuracy is suitably sufficient for most practical applications.

VI. Effect of Altitude on the ROM

One of the biggest sources of potential computational savings come from the ability of the ROM to be created at one altitude and then reused for other altitudes; so long as the Mach number remains constant. To achieve this, is necessary to modify the stepdown response by applying Eq. (27):

$$t_{new} = t_{old}\left(\frac{U_{old}}{U_{new}}\right) \tag{27}$$

This modification corrects the rate at which the model responds to the gust by slowing down (if the original velocity is greater than the new velocity) the rate at which it effectively passed through the sharp-edged gust used to create the stepdown response. However, this modification will in turn reduce (if the original velocity is greater than the new velocity) the amplitude of the response, and so it is necessary to apply Eq. (28) to correct for this:

$$U_{dt_{new}} = U_{dt_{old}} \left(\frac{U_{old}}{U_{new}} \right) \tag{28}$$

Together, these corrections allow for the ROM to be built at one altitude and then used for any altitude so long as the Mach number remains constant; this in turn can drastically reduce the computational cost if multiple altitudes are required per Mach number. To test the corrections, flight point 4 from Table 1 was run again, with the ROM built for flight point 2.

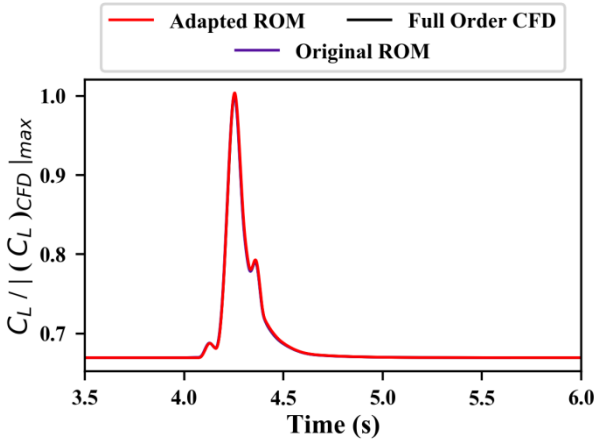

Figure 48. C_L response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 1.

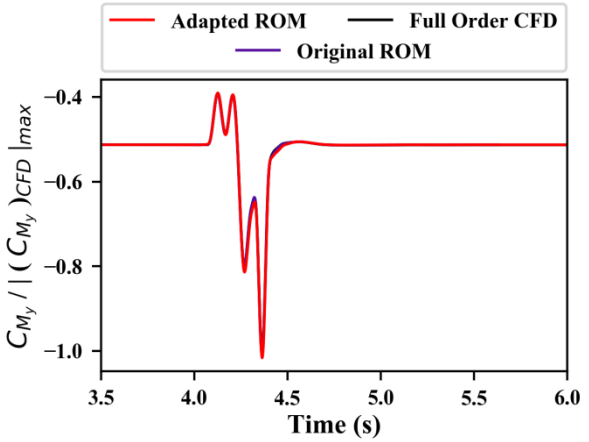

Figure 49. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 1.

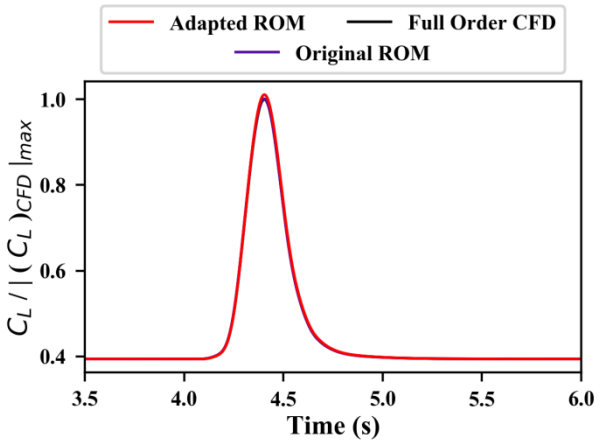

Figure 50. C_L response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 2.

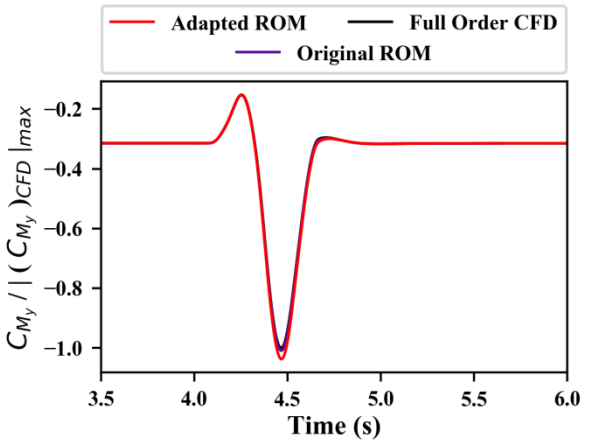

Figure 51. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 2.

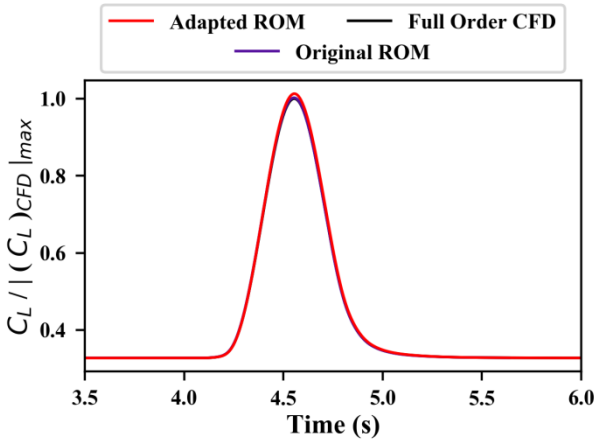

Figure 52. C_L response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 3.

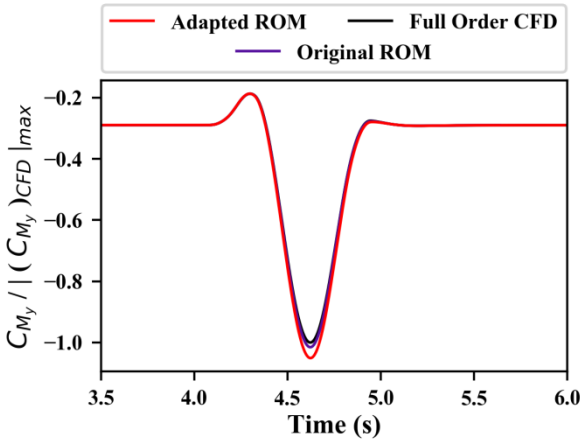

Figure 53. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 3.

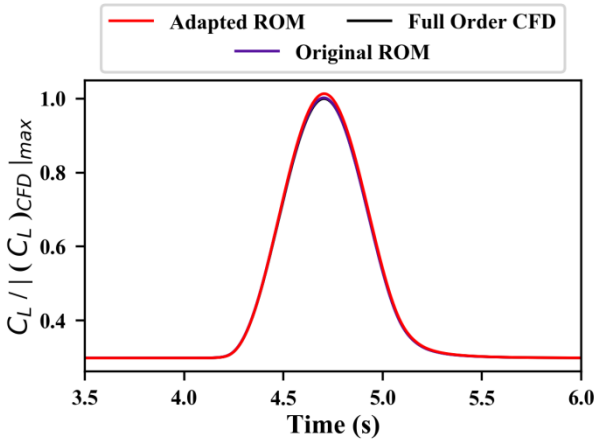

Figure 54. C_L response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 4.

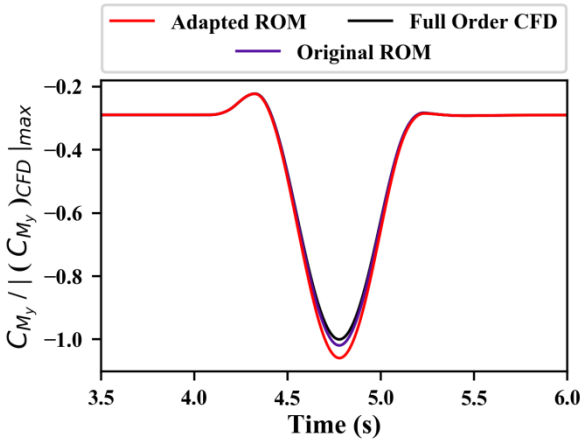

Figure 55. C_{My} response for a viscous, aeroelastic half-aircraft model for flight point 4, gust case 4.

It can be seen in Fig. 48 to Fig. 55 that whilst there is some degradation in accuracy (notably at the peak response), which gets worse with increasing gust length, overall the accuracy is largely maintained. However, by reutilising the flight point 2 ROM, there was negligible computational cost in producing these results, highlighting the potential use of the ROM in early design work where resources are limited and designs are changing on a semi-regular basis.

VII. Application of ROM to Flight Mechanics Case

The ROM is designed to include as much information in the output as it obtains during its creation. That is to say, if (for example) the ROM is built using an aeroelastic model, the output will contain these characteristics. Therefore if the ROM is built with flight mechanics included, it should give an output that includes this as well. To test it, the ROM is built for a simple model consisting purely of a wing & empennage, and then simulated in an inviscid flow, but including aeroelastic effects. The model is trimmed prior to the impact of the gust, using the angle of attack and the elevator as the control variables. As the gust passes over the aircraft, the elevator deflection is kept constant, but the angle of attack is free to change. Whilst work in this area is still ongoing, the results (Fig. 56 to Fig. 59) look promising.

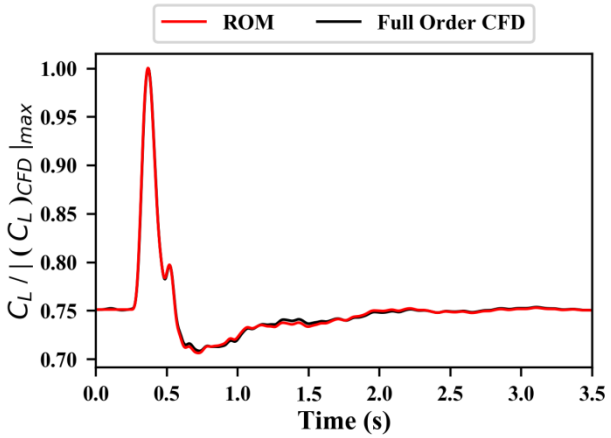

Figure 56. C_L response for an inviscid, aeroelastic simple aircraft model with basic flight mechanics for a 18.288 meter long gust, with a magnitude of 6.11 meters per second.

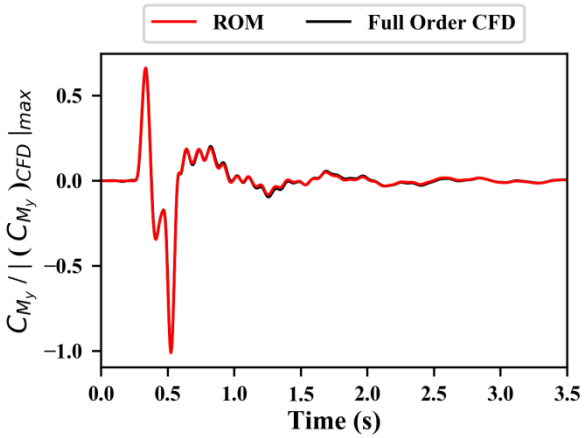

Figure 57. C_{My} response for an inviscid, aeroelastic simple aircraft model with basic flight mechanics for a 18.288 meter long gust, with a magnitude of 6.11 meters per second.

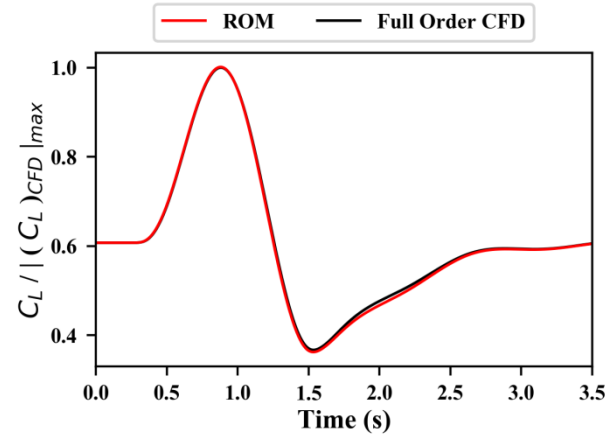

Figure 58. C_L response for an inviscid, aeroelastic simple aircraft model with basic flight mechanics for a 213.36 meter long gust, with a magnitude of 6.11 meters per second.

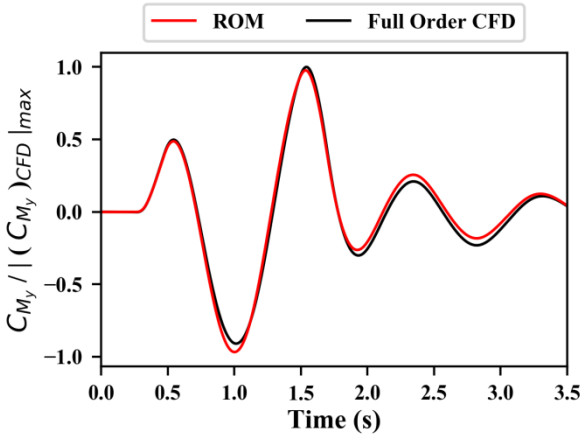

Figure 59. C_{My} response for an inviscid, aeroelastic simple aircraft model with basic flight mechanics for a 213.36 meter long gust, with a magnitude of 6.11 meters per second.

VIII. Conclusion

It has been shown that the ROM shows strong promise for use in the design of aircraft. It's high accuracy (relative to full order CFD results) coupled with its low computational resources make it ideally suited for early design phases, when designs change frequently and computational resources are at a premium, but it is also a useful tool throughout design process.

The modification from building the Hankel matrix using 'step-up' data, to stepdown, combined with the improvements to the stabilisation of the ROM open the possibility to the sharp-edge gust being ended earlier; reducing the time steps required and therefore the computational cost associated with building the ROM. Whilst this is an area of ongoing development, particularly the stabilisation of the ROM, the results shown already highlight a marked improvement from the previous methods used.

The results from the half-aircraft, aeroelastic tests show extremely high accuracy for smaller gusts & at lower altitudes, and although it was shown that the increasing non-linearities associated with larger gusts & higher altitudes did cause a reduction in accuracy, the results were still very close to the full order CFD. These inaccuracies did grow when adapting a ROM built at sea level for use at other altitudes (with the same Mach number), but again the results showed a good general match to the full order CFD results and the error was sufficiently small as to not

be of concern; particularly when put into the context of the computational savings offered by only needing to build one ROM per Mach number.

Whilst the results from the basic flight mechanics tests represent the early stages of this part of the ROM's development, the results shown are highly encouraging and demonstrate that the ROM is capable of capturing in its output whatever information (aeroelastic effects, flight mechanics, etc.) is included when building it. This in turn highlights the ROM's versatile nature and thus it's potential use in the design of aircraft.

Acknowledgments

The authors would like to thank RCUK for the funding which has made the work laid forth in this paper possible. The authors would also like to thank AGI for their support, both financial and otherwise, which has been invaluable; in particular, S. P. I. Williams would like to thank Martin Herring and Dr. John Pattinson for their ongoing support.

References

[1] Wright, J., and Cooper, J., *Introduction to Aircraft Aeroelasticity and Loads*, Chichester, West Sussex: Wiley, 2015.

[2] European Aviation Safety Agency, "Certification Specificaitons and Acceptable Means of Compliance for Large Aeroplanes - CS-25" Available: https://www.easa.europa.eu/system/files/dfu/CS-25 Amendment 18_0.pdf.

[3] Mai, H., Neumann, J., and Hennings, H., "Gust response: a validation experiment and preliminary numcerical simulations," *15th International Forum on Aeroelasticity and Structural Dynamics*, 2011, pp. 1–20.

[4] Wales, C., Jones, D., and Gaitonde, A., "Reduced order modelling for aeroelastic aerofoil response to a gust," *51st AIAA Aerospace Sciences Meeting*, 2013, pp. 1–16.

[5] Wales, C., Jones, D., and Gaitonde, A., "Prescribed Velocity Method for Simulation of Aerofoil Gust Responses," *Journal of Aircraft*, vol. 52, 2015, pp. 64–76.

[6] Williams, S. P., Jones, D., Gaitonde, A., Wales, C., and Huntley, S. J., "Application of Reduced Order Models in Aircraft Gust Response Studies," *46th AIAA Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2016.

[7] Juang, J.-N., and Suzuki, H., "An Eigensystem Realization Algorithm in Frequency Domain for modal parameter identification," *Journal of vibration, acoustics, stress, and reliability in design*, vol. 110, 1986, pp. 24–29.

[8] Silva, W. A., "Application of Nonlinear Systems Theory to Transonic Unsteady Aerodynamic Responses," *Journal of Aircraft*, vol. 30, 1993, pp. 660–668.

[9] Silva, W. A., "Extension of Nonlinear systems Theory to General Frequency Unsteady Transonic Aerodynamic Responses," *34th AIAA Structures, Structural Dynamics, and Materials Conference*, Reston, VA.: AIAA, 1993, pp. 2490–2503.

[10] Silva, W. A., "Identification of Linear and Nonlinear Aerodynamic Impulse Reponses Using Digital Filter Techniques," *AIAA Atmospheric Flight Mechanics Conference*, Reston, VA.: AIAA, 1997, pp. 584–597.

[11] Ma, Z., Ahuja, S., and Rowley, C. W., "Reduced-order models for control of fluids using the eigensystem realization algorithm," *Theoretical and Computational Fluid Dynamics*, vol. 25, 2011, pp. 233–247.

[12] Wales, C., Gaitonde, A., and Jones, D., "Reduced-order modeling of gust responses," *Journal of Aircraft*, Dec. 2016.

[13] Wales, C. J. A., Gaitonde, A. L., and Jones, D. P., "Stabilisation of Reduced Order Models via Restarting," *International Journal for Numerical Methods in Fluids*, vol. 73, 2013, pp. 578–599.

[14] McKelvey, T., Akçay, H., and Ljung, L., "Subspace-based multivariable system identification from frequency response data," *IEEE Transactions on Automatic Control*, vol. 41, 1996, pp. 960–979.